GEORGIA INSTITUTE OF TECHNOLOGY OFFICE OF CONTRACT ADMINISTRATION

# PROJECT ADMINISTRATION DATA SHEET

[X] ORIGINAL [ ] REVISION NO. ______

Project No. E-24-648 (R6135-0A0) GTRC/~~GIT~~ DATE 5 / 19 / 86

Project Director: Dr. H. D. Ratliff School/~~Lab~~ ISyE

Sponsor: Department of the Navy, Office of Naval Research
Arlington, VA

Type Agreement: SFRC Contract No. N00014-86-K-0173 and Mod. P00001

Award Period: From 2/1/86 To 1/31/87 (Performance) 3/31/87 (Reports)

| Sponsor Amount: | This Change | Total to Date |
|---|---|---|
| Estimated: | $ | $ 190,000 |
| Funded: | $ 190,000 | $ 190,000 |

Cost Sharing Amount: $ 12,351 Cost Sharing No: E-24-353 (F6135-0A0)

Title: Production and Distribution Research Center

## ADMINISTRATIVE DATA

OCA Contact E. Faith Gleason

1) Sponsor Technical Contact:

Dr. Neil D. Glassman
Office of Naval Research
800 North Quincy Street
Code ~~1111~~ N00014
Arlington, VA 22217-5000
(202) 696-4310

2) Sponsor Admin/Contractual Matters:

ONR RR
206 O'Keefe Building
Georgia Institute of Technology
Atlanta, Georgia 30332-0490
347-4381

Defense Priority Rating: ______ Military Security Classification: Unclassified

(or) Company/Industrial Proprietary: ______

## RESTRICTIONS

See Attached Gov't Supplemental Information Sheet for Additional Requirements.

Travel: Foreign travel must have prior approval – Contact OCA in each case. Domestic travel requires sponsor approval where total will exceed greater of $500 or 125% of approved proposal budget category.

Equipment: Title vests with GIT if less than $5,000 and which has been authorized by Government Contracting Officer

## COMMENTS:

Follow-on Contract to E-24-617:N00014-83-K-0147

MAY 1986 RECEIVED SUPPORT SERVICES OCA

## COPIES TO:

SPONSOR'S I. D. NO. 02.103.000.86.011

Project Director
Research Administrative Network
Research Property Management
Accounting

Procurement/EES Supply Services
Research Security Services
Reports Coordinator (OCA)
Research Communications (2)

GTRC
Library
Project File
Other A. Jones/Legal

SIP 5452

# GEORGIA INSTITUTE OF TECHNOLOGY
# OFFICE OF CONTRACT ADMINISTRATION

## NOTICE OF PROJECT CLOSEOUT

Date 9/29/89

Project No. E-24-648 Center No. R6135-0A0

Project Director H. D. Ratliff School/Lab ISyE

Sponsor Navy

Contract/Grant No. N00014-86-K-0173 GTRC XX GIT ___

Prime Contract No. N/A

Title Production and Distribution Center

Effective Completion Date 1/31/89 (Performance) 3/31/89 (Reports)

Closeout Actions Required:

- X None
- ___ Final Invoice or Copy of Last Invoice
- ___ Final Report of Inventions and/or Subcontracts
- ___ Government Property Inventory & Related Certificate
- ___ Classified Material Certificate
- ___ Release and Assignment
- ___ Other ___

Includes Subproject No(s). ___

Subproject Under Main Project No. ___

Continues Project No. ___ Continued by Project No. ___

Distribution:

- X Project Director
- X Administrative Network
- X Accounting
- X Procurement/GTRI Supply Services
- X Research Property Management
- X Research Security Services
- X Reports Coordinator (OCA)
- X GTRC
- X Project File
- X Contract Support Division (OCA)
- ___ Other ___

## III. PROPOSED RESEARCH

We propose to continue the basic focus of our research on the development of mathematical constructs, models, and algorithms to aid in the design, operation, and control of systems for moving and storing material.

The central theme of our research involves the identification of spatial and combinatorial characteristics which allow reductions in the dimensionality of the problems. These reductions in dimensionality take a variety of forms. For order picking problems, the restricted travel within the warehouses frequently leads to efficient optimum solution techniques. For delivery problems, the space filling curve concept allows the problem to be mapped into a problem on a line. This cannot be solved optimally, but does lead to heuristics with provable bounds. For the interactive optimization approaches to layout and scheduling, optimization models are able to handle certain aspects of the problem, thus leaving the human designer with a much simpler problem to address. We intend to extend and build on our previous research in each of these areas. PDRC Reports 86-12 and 86-11 reflect our current research in order picking and interactive scheduling. PDRC Report 86-09 gives new results related to network aggregation. A PDRC report reflecting new results in interactive layout is currently being written.

During the past year we have also began to examine similar concepts in physical aggregation problems (e.g., product assembly, and consolidation/sorting of material for shipment), decentralized control of automatic guided vehicles, two dimensional packing problems, and partitioning of composite optimization problems. These problems are discussed in more detail below.

## Product Assembly

Assembly of a product from its components is essentially an aggregation process which takes components and aggregates them. To complete the assembly in the minimum elapsed time requires that tasks be performed in parallel whenever possible. Since a decision on an assembly sequence requires information regarding feasible subassemblies, we require a compact representation of the feasible assembly region. One compact representation of the feasible subassembly region of a product, is obtained by constructing a tree graph. Every node in the graph corresponds to a component in the product. An edge in the graph implies an assembly process. Complete assembly of the product requires all assembly operations implied by the edges to be performed. If every connected component of the tree is considered to be a feasible subassembly of components, then all possible subassembled structures of the product are also connected components in the graph. The graph representation also implies that not more than two subassemblies may be involved in an assembly process. Furthermore, in the general case there may be different assembly times depending on the components being assembled. This can be represented as weights on the edges of the graph.

We have analyzed the problem of minimum time assembly of products whose assembly can be represented by a tree structure where all assembly operations are assumed to take unit time. When two components are assembled, the feasible region of the components not involved in the assembly and the subassembly formed are represented as follows. The edge corresponding to the assembly operation is "collapsed" and the components involved in the assembly operation are combined to form a super node. The super node is connected to those components in the rest of the graph that each of the components involved in the assembly were connected to. All feasible assembly structures in the new set of available components are represented by the new graph created. Furthermore, the new graph created is an 'aggregation' of the original graph structure.

When an assembly operation is being performed, components not involved in the assembly process (some of the components may be subassembled components) may also be assembled in parallel operations. Thus if an operation were restricted to involve at most two components, the set of feasible assembly operations that can be done in parallel at any time could be represented by a 'matching' of edges in the corresponding graph. Since a matching identifies a set of edges that are disjoint (i.e. no two of them are incident to the same node), it also identifies sets of feasible parallel assembly operations.

Since each step in the assembly process aggregates components, performing only feasible assembly operations, the product (fully assembled) would be represented by a graph that is a single super node. The minimum time assembly problem can therefore be described as finding the minimum number of steps, each step involving a matching on the aggregated graph from the earlier step, required to collapse the graph into a single node. We observe that performing a maximum matching of assembly operations at each step may not yield the minimum number of steps.

The assembly operations and feasible matchings can be described in terms of the original graph itself as edge labeling operations that satisfy the property that between two edges of the same label i, there is at least one edge with a label $j > i$. This problem represents a constrained version of the edge coloring problem on a tree graph. Since edge coloring on a tree can be solved by a linear time algorithm, an interesting research question is whether the constrained version of the problem is solvable quickly.

The same problem can also be described as that of accepting a tree graph G as input and creating a binary tree T of minimum height such that the vertices of G are the leaf nodes of T and the edges of G are the internal nodes of T. In this framework, the problem is related to the edge separator tree of a tree, described by Leighton (1982) in the context of circuit layout.

Some of the results obtained regarding this problem are as follows.

(1) For special tree structures such as complete binary trees, chains of stars, stars, chains, star of chains etc. optimal algorithms have been obtained.

(2) For the weighted chain structure, a polynomial algorithm is developed that provides the minimum time assembly of the components.

(3) Given a set of optimally labeled subtrees all connected to a single node, we can label the new graph (consisting of the original subtrees and the edges linking it to the additional node) by using no more than one label over the optimal i.e.

$HEUR( i + T_1 + \cdots + T_k) <= OPT( i + T_1 + \cdots + T_k) + 1$

(4) For any arbitrary tree structure, we can use the node separator tree of Lipton and Tarjan (1979), which is guaranteed to be of height <= O(log n), to develop an algorithm with a guaranteed constant worst case bound.

However, some important research issues to be resolved are

(1) Is there an efficient algorithm to solve the problem on a general tree structure ?

(2) Are there other commonly occurring special structures for which this problem is solvable ?

(3) Can this notion be generalized to arbitrary graph structures (i.e. including graphs with cycles). Can some strategy be devised that identifies a tree subgraph and applies the procedures described above.

<u>Consolidation and Sorting in Distribution Systems</u>

In material movement systems where packages are picked up at multiple origins and delivered to multiple destinations, aggregation of packages in regions for re-distribution is a common strategy used. Decisions have to be made regarding the regions formed, re-distribution strategies, sorting mechanisms etc. Typically binding constraints are that the volume handled by each of the regions formed be reasonable, and that the time to deliver a package from any one point to another be guaranteed to be less than a certain input value. Other constraints include availability of resources i.e. number of resources, capacity etc. In general we have a complete graph structure on the nodes (being points of origin of material), and also information regarding the flow from point i to point j.

Consider a tree structure linking the points where packages are to be delivered. Also, suppose there is a parameter associated with each edge which can be used to separate the node set into two connected subtrees. This implies that each destination on n nodes has associated with it an n-1 size vector of 0 or 1 with respect to the parameters on each of the n-1 edges. Now suppose all the material to be delivered is accumulated at a point. The minimum height binary tree discussed in the earlier section would also provide the smallest number of sorting steps required to separate the packages into individual entities, where a sorting step accepts a group of packages and separates them into two groups based on the value of a parameter being 0 or 1.

Consider a sorting mechanism which has a complete binary tree structure. A parameter can be set on each of the internal nodes. The whole set of packages is introduced at the root node and get separated out into groups based on the parameter settings at the internal comparator nodes. Now, if the minimum height of the binary tree required to separate the entities is lower than the sorter height, then the sorting can be performed in using one sorter. However, if we are given a fixed height binary sorter, we need to decide on the location and inputs to various sorters, while minimizing the number of such sorters used. Even if we have a minimum height binary tree, we may not have a layout of the nodes that minimizes the number of sorters used. However, for a given tree layout, we provide a linear time algorithm that produces the minimum number of sorters necessary and their location. Therefore an issue to be resolved is "can we develop an algorithm to decide on the minimum number of sorters and their locations on the tree structure?".

Consider the basic problem of locating sorting stations that service a group of nodes that are connected. The sorting station essentially operates in two phases. In the first phase, the packages originating from and destined for nodes in a subtree are sorted down to their individual levels. However, packages headed from this subtree region to other subtree regions are represented as aggregate groups, one for each of the regions. In the next phase, all these aggregate groups are re-distributed and all packages addressed to nodes in a region are accumulated. Then, this aggregate set is sorted down to its individual entities. Therefore, the sorting station capacity would influence the total flow exchanged within a region and also entering a region from other regions. We show that even on a tree structure where flows are exchanged only between adjacent nodes, the problem is NP-complete Garey and Johnson (1979), by reduction from the two partition problem.

The next problem is that of re-distribution given a constraint on the total flow into each subtree. This implies that for each node we compute the total flow into it from all other nodes and that all the sorting into detailed entities is done after all the re-distribution is completed. However, the time it takes for every package to get from one point to another is constrained. If this constraint were relaxed, then Hadlock's (1974) algorithm would provide a partition of the subtree into a minimum number of stations. Since re-distribution of the packages can be considered to be proportional to the distances on the cut edges between subtrees (the tree structure implies that there is at most one such edge between any pair of subtrees), the problem is to partition the tree into subtrees of capacity <= W while the sum of the flows on the cut edges is <= T. This problem is also NP-complete by reduction from the knapsack problem.

Therefore both the load balancing problem for sorting stations and the time constraint on the package delivery make the problem NP-complete. Hence, we consider cases where only one or the other constraint applies. Consider the time constraint on the flows between points. The flow process consists of accumulating packages within each subtree region, then re-distributing packages across regions and finally sorting them down to their individual levels. If a unit of collection resource were available at each subtree and at the higher inter-regional level, the problem becomes that of dividing a tree into regions so that the sum of the edge weights in each region is constrained <= W and the sum of the cut edge weights is constrained to be <= T. This problem again is NP-complete by two-partition.

An alternative system would be for packages to move independently to the center of their corresponding subtree regions (radius of regions being constrained) and for the sum of the cut edge weights to be <= T. The complexity of this problem is being examined. Therefore some of the issues to be resolved are

(2) Can we develop an algorithm to divide a tree into connected subtrees of radius <= r, the sum of the cut edge weights being <= T ?

(3) Can we develop algorithms with provable performance measures for some of the NP-complete problems discussed in this section.

(4) Can we identify 'good' tree structures on the original graph which can be evaluated with respect to various parameters such as minimum number of sorters, minimum time to deliver packages across points etc.

Decentralized Control of Automatic Guided Vehicles (AGV's).

The control of AGV's is proving to be a very good and practical arena in which to test our ideas on decentralized, locally autonomous optimization. The automated factory or warehouse which uses AGV's can be seen as a hierarchical, real time environment in which decisions must be made locally, for speed and reliability of the system. We are studying how to decentralize optimization/control and then how to integrate the responses of the subsytems.

We have been helped in this by cooperation from Litton Industries, a major manufacturer of AGV's. With them we have developed a classification scheme for AGV systems and control strategies, and for each system we are establishing performance bounds for different control strategies. Our results so far indicate that even simple decentralized control heuristics can be guaranteed to work surprisingly well.

We have studied the "Busy Heuristic" (discussed in the previous proposal), and have additionally determined that, while it is guaranteed to perform exceptionally well under the criterion of "latest delivery time", it is capable of poor performance for average throughput. (This is to be expected since the Busy Heuristic is the most myopic practical control strategy.) To understand how to ensure good throughput, we have analyzed the "Soonest Delivery Heuristic", which, when there is space available aboard the AGV, next picks up the item which is closest to its destination. Notice that this requires more look-ahead than does the Busy Heuristic, but it is nevertheless sufficiently decentralized to allow different AGV's to operate independently of each other; they need query only the load/unload stations, and not the other vehicles.

We have proven that, under the Soonest Delivery Heuristic the following is true.

1. The time of the last delivery of a tote will be no more than twice the optimum time.

2. The AVERAGE time of delivery of all totes will be no more than twice the optimum.

In this analysis, "optimum" is unrealistically small. It is calculated for the ideal case in which the AGV can randomly access any tote at a load/unload station (rather than observe queue order), and totes can be temporarily deposited at intermediate stations, to be delivered later.

Now we are studying "priority-based" heuristics, wherein totes of high priority are given first consideration for delivery.

We are also studying the effects of network topology on the performance of the scheduling rules. Here our results are facilitated by the fact that most real AGV networks are "series-parallel". Since such networks have simple recursive structure, we can assign part of the routing control functions to components of the network. Then distinguished nodes of the network can work with the simple AGV control to achieve high quality routing, without centralized computation. For example, a portion of a typical AGV network includes multiple lanes for pickup/deliveries. In this component of the AGV network, install a linked pair of autonomous controllers at entry point A/exit point B. We assign these controllers responsibility for only and exactly the paths between A and B. Now these controllers can work with, say, the Busy Heuristic to route the AGV appropriately: at A the AGV receives instructions as to which lane to travel; through the lane, the AGV uses its local "intelligence" to determine pickups and deliveries; at B the AGV reports what pickups were made; B relays this information back to A, so that A can appropriately route the next AGV. We have preliminary results on the performance of such configurations that show that worst-case performance can be bounded by small factor that depends on the complexity of the structure of the AGV network (which in practice is small).

This research will be presented at an invited presentation to the ORSA/TIMS Joint National Meeting in Miami, 1986. It is will be reported on more thoroughly in the doctoral dissertation of Wang Lim (expected date of graduation: summer 1987).

## Two-dimensional Packing

Spatial packing problems are a good test area in which to explore issues of autonomous optimization because spatial problems can be decomposed simply and naturally. Typically, packing in one region affects only slightly the packing in more distant regions. "Shelf heuristics" attempt to exploit this directly, by partitioning the region to be packed into "shelves" which then can be packed independently and in parallel. A centralized intelligence is necessary only to assign each item to be packed to the appropriate shelf.

A further advantage of shelf heuristics is that they result in subproblem simplification: each shelf can be packed well by simple procedures since the allocator has screened the items sent there to be packed. For example, each shelf can be packed as a simple 1-dimensional packing problem, and still yield good results. In addition, the simplicity of the subproblem enables the packing to be computed "on-line", since the entire problem need not be known in advance. The partitioning ensures that there will be little significant interaction among the items in the shelves (for large problems).

Thus, when the problem is decomposed appropriately, items allocated appropriately, and subproblems solved appropriately, the problem can be solved well at the highest level. was first studied by B. Baker, E. G. Coffman and R. L. Rivest (1980). They formalized the problem as follows: Given a bin with fixed width and unbounded height, and a list of rectangles ("boxes"), pack the boxes into the bin so that no two boxes overlap and so that the height to which the bin is filled is as small as possible. We assume that the boxes are "oriented", each having a specified side that must be parallel to the bottom of the bin. 2-dimensional bin-packing has widespread applications, including stock cutting, warehouse storage, scheduling with resource constraints, etc. Because of the intractability of packing, research has concentrated on analyses of heuristics. The heuristics can be classified into two main types:

1. Off-line algorithms are allowed to defer the packing of boxes and rearrange boxes that have already been packed. This corresponds to the algorithm having complete knowledge in advance of the problem, so that the problem can be solved completely before the solution is implemented.

2. On-line algorithms must pack boxes in the same order as they are listed. This is the case when the algorithm must solve the problem and implement the solution even as the problem is being revealed. For example, imagine the boxes to be packed arriving over time. An on-line algorithm must pack the boxes as they arrive, and without knowing anything about future arrivals.

On-line heuristics are less accurate in general than off-line heuristics since they must work with less information. Also they are constrained to pack boxes according to the sequence in which they appear on the list.

As is generally the case, worst-case bounds are more easily available than expected case bounds. Worst-case bounds are either

1. asymptotic, i.e. of the form Heuristic $<$ a * OPT + b, or

2. absolute, i.e. of the form Heuristic < a * OPT.

Absolute bounds are generally not as good as asymptotic bounds, since the usual heuristics can typically be fooled badly on very small, skewed problems. This is because most heuristics do some sort of partitioning, and the problems must be "large enough" so that the partitions are fully used.

We generalize the computer science model of bin-packing to this: Given a set of storage areas of specified shapes and dimensions, pack a list of boxes into the storage areas so as to waste as little space as possible. Our results are for a special type of on-line heuristic known as a "shelf" heuristic. The idea is to pack boxes onto "shelves", which are constructed as needed. A central issue is to decide what size shelves to use. After it is constructed, each shelf can be treated as a bin of one less dimension. (For example, in 2-dimensional bin-packing a shelf is simply a 1-dimensional bin, where the single dimension is the width of the shelf). This enables us to reduce higher dimensional packing problems to simpler, lower dimensional packing problems - however, at some cost in accuracy of the solution. Our results suggest that in practice this trade-off can frequently be worth making. Our results are strong in that we compare heuristic performance with the unrealistically good solution possible when boxes are completely deformable, and so can be packed with absolutely no wasted space (as if we were packing water rather than boxes). This is a lower bound even on the very best possible when the future is completely known and we are willing to devote whatever computing power necessary to take advantage of that knowledge. Our results so far include:

1. An analysis of how the set of shelf sizes affects the performance of Baker and Schwartz's heuristic. A result of this is a better set of shelf sizes that lead to improved bounds. Furthermore we show lower bounds on the worst-case performance of any shelf heuristic, independent of the specific shelf sizes.

2. Average-case performance analysis for Baker and Schwartz's shelf heuristic. The result shows how the shelf sizes can be chosen to achieve good worst-case performance but at the cost of diminished expected performance, or to achieve good expected performance at the cost of diminished worst-case performance. Thus worst-case and expected performance can be balanced in advance by the user.

3. Analysis of the shelf heuristic in the special cases of when

a. the storage area consists of a set of square bins (rather than a single open-ended bin).

b. when the largest and smallest dimensions of the boxes are known a priori. (Considerable better bounds are possible when the dimensions of the smallest box are not "too" much smaller than those of the largest box to be packed. This would be the case in a warehouse.)

4. How to determine the best shelf sizes when only a fixed, finite number of shelf sizes are allowed and a probability distribution is known for the dimensions of arriving boxes.

Results for when the storage area is either

a. a single open-ended bin, or

b. finite set of squares.

5. How to determine the optimal shelf sizes for the "dynamic bin-packing problem", in which boxes enter the system, are packed and remain in storage for some time, and then leave the system in the following cases:

a. we know a priori probabilistic information about arrivals and departures of boxes;

b. we have no a priori knowledge of arrivals and departures of boxes.

6. Worst-case analysis of a "shelf and floor" heuristic for 3- dimensional packing when the storage space is either

a. a single open-ended bin, or

b. a set of cubes.

In addition to the above results, we have a number of partial results that we expect to resolve. We are also broadening the problem to incorporate physical constraints, such as requiring "stable" packings (so boxes will not tip over), even distribution of weight, etc.

This work will be reported on in the doctoral dissertation of Zhang Jixian (expected date of graduation: August 1987).

Voting Procedures

We have also begun studying how to combine subproblem solutions so that their quality is preserved and transmitted to higher levels in the hierarchy of problem components. In highly decomposable systems, such as the packing problems above, integrating subproblem solutions is trivial. For less decomposable systems, the issues are considerable more complex. In such systems, we can imagine each autonomous optimizer to produce a "preference ordering" of alternative solutions. Then the higher level controller is like a "voting procedure", which intelligently integrate these subproblem solutions to produce a group decision. What sort of "voting procedures" should be used, and when can they be guaranteed to work acceptably?

In principle such procedures are susceptible to the anomalous behavior of known to apply to social voting systems; but our procedures need not be constrained to satisfy criteria of "fairness" or "social rationality". Given this extra leeway, can we design functional procedures to integrate the recommendations of the autonomous optimizers?

We have identified some paradigms in which this is possible. For example, the Stable Matching problem asks to pair 2n nodes so that no 2 nodes prefer each other to their current partners. In a sense each node is an autonomous decision maker who must accommodate the preferences of all others. In general there is no solution to this problem. However, given a structure on the preferences which plausibly models the psychology of a human formulator of criteria, a stable matching always exists, is unique, and can be constructed quickly under the auspices of a very weakly centralized algorithm which simply coordinates the self interest of the individual nodes. Furthermore, since the algorithm relies so much on the self interest of the nodes, it is highly parallelizable. (This has been reported on in "Stable matching with preferences derived from a psychological model" by John J. Bartholdi, III and Michael A. Trick, to appear in Operations Research Letters.) We expect more of this type of result, which give non-trivial ways of integrating the preferences of subsystems, but while preserving a high degree of decentralization.

We remark parenthetically that this line of thought has led to an unexpected result that is of considerable interest for social decision-making: we have proved that a clearly defined, historical voting scheme has the property that it can be "hard" to determine the winner! (Specifically, to determine the score of any candidate is NP-complete, and to determine the winner is $D^p$-equivalent.) This result is apparently the first to introduce computational complexity into social decision-making; and, moreover, is interesting for historians of voting procedures. This will be reported on in "The Complexity of Voting" by John J. Bartholdi, III, Craig A. Tovey, and Michael A. Trick.

Another approach to integrating subproblem solutions is to allow the system to "evolve" a procedure. For example, one might simulate the system, measure performance, and feed this back to the local decision makers, who are provided with the ability to mutate in response to the feedback. Given the right evolutionary pressure, the system will presumably evolve toward efficient performance. We have established just this sort of behavior for a system based on iterated plays of a formalized game, reported on in "More on the evolution of cooperation" by John J. Bartholdi, III, C. Allen Butler, and Michael A. Trick, Journal of Conflict Resolution, Vol. 30 No. 1, March 1986, pp. 129-140.

## Solving Optimization Problems by Variable Splitting

Many difficult optimization problems contain large subproblems that, if solved independently, are tractable. The difficulty arises in the relationships among the subproblems and, possibly, difficult extra constraints. The deployment planning problem is such a problem. It consists of a transportation configuration problem and a movement requirement assignment problem. Each of these problems considered alone (i.e. with the variables in the other problem fixed) is a minimum cost network flow problem. Taken together, the problem is a linear programming problem, not a network flow problem. Another example is the Travelling Salesman Problem (TSP). Here the goal is to find a minimum length route that visits each of a set of cities exactly once. One subproblem of the TSP is to find a set of links between cities (edges) so that each city is adjacent to two edges. Another subproblem is to find a set of edges that joins all of the cities. Each subproblem is easy to solve alone (the first by matching techniques, the second by spanning tree methods), but together the TSP is notoriously difficult to solve.

In most cases where there are large, easily solved subproblems, techniques are available to exploit the structure. In some cases (like the deployment problem) these techniques are more effective methods for finding the optimal solution. In other cases (like the TSP) exploiting the structure of the subproblem is not guaranteed to provide the optimal solution to the original problem. However, good estimates of the optimal cost are given. These estimates can be incorporated in other techniques (like Branch and Bound), enhancing the search for the optimal solution.

One general method for attacking these problems is Lagrangian relaxation (Geoffrion (1974)). Lagrangian relaxation can be applied where there is a large, obvious, portion with exploitable structure and a small number of complicating constraints. In this case, the complicating constraints are multiplied by some cost, representing the cost of removing that constraint, and placed into the objective. In this case, some of the cost is allocated to the subproblem and the rest is allocated to the relaxed constraint, in the form of a constant.

Lagrangian relaxation has proved effective in a wide variety of practical problems (Fisher (1985)). One fruitful area of research has been methods to expand the applicability of Lagrangian relaxation. For instance, Lagrangian relaxation cannot be applied directly to the TSP formulation given above because both subproblems have a enormous number of constraints, precluding placing either set into the objective.

Variable splitting is a promising, general technique for bringing out exploitable structure when it is hidden in problems. In variable splitting, each variable is replaced by two or more new variables. Each constraint involving the old variable receives one of the new variables instead. If all of the new variables are forced to take on the same value, then the new problem is exactly the same as the old problem. If variable splitting is done cleverly then a problem can be created that is amenable to Lagrangian relaxation or some other technique to exploit special structure.

For instance, the TSP can be written as the following integer program:

$$
\begin{aligned}
&\text{Minimize} \quad c\,x \\
&\text{Subject to} \\
&\qquad\qquad A_1\, x \le b_1 \\
&\qquad\qquad A_2\, x \le b_2 \\
&\qquad\qquad\quad x \ge 0,\ \text{integer},
\end{aligned}
$$

where the $A_1$ constraints represent the requirement that two edges hit each city and the $A_2$ constraints force the tour to be connected. Each variable $x_i$ can be replaced with two variables $y_i$ and $z_i$. Every occurrence of $x_i$ in $A_1$ is replaced with $y_i$ while those in $A_2$ are replaced with $z_i$. Constraints forcing $y_i$ to equal $z_i$ are added resulting in the problem:

$$
\begin{array}{llcl}
\text{Minimize} & c\,y & & \\
\text{Subject to} & & & \\
& A_1\, y \le b_1 & & \\
& & A_2 & z \le b_2 \\
& y & - & z = 0 \\
& y, & & z \ge 0,\ \text{integer}.
\end{array}
$$

Now the $y-z = 0$ constraints can be relaxed by Lagrangian relaxation. Note that the number of constraints relaxed depends only on the number of variables, not the number of constraints in $A^1$ or $A_2$. Also, the problems in y and z are completely separate, so the structure of $A_1$ and $A_2$ can be exploited.

The problem created by splitting the variables and relaxing the constraints forcing the new variables to be equal is called the variable splitting problem.

Some questions that we have addressed in the past year are:

1) What is the relationship between variable splitting and other relaxations, such as linear programming and Lagrangian relaxation on the original problem?

2) Does variable splitting give any extra information about the original optimization problem beyond the estimate of the optimal objective function value?

3) How can the problem that results from performing Lagrangian relaxation after variable splitting be solved?

4) What sort of problems are amenable to variable splitting?

The key to comparing variable splitting with other relaxations is to see relaxations as optimizing over an expanded set of feasible points. The original optimization problem is concerned with optimizing over the smallest feasible region. The linear programming relaxation optimizes over a region at least as large. We can show that no matter how the variables are split, the region optimized over is no larger than the linear programming region. This implies that variable splitting is at least as good as linear programming. In cases where the problem has a natural Lagrangian Relaxation, creating one set of variables for the specially structured subproblem and another for the relaxed constraints is a natural variable split. In this case, the region for the variable split problem is contained in the region for the natural Lagrangian relaxation, so variable splitting is at least as good as Lagrangian relaxation.

This sort of analysis also shows when the relaxations will give the same results: when the regions are the same. A key concept is the integrality property. A problem has the integrality property when dropping the integrality restrictions does not affect the optimization property. Some examples of problems with integrality property are network flow problems and bipartite matching problems. If the problem created by variable splitting (relaxing the constraints that force the new variable to equal each other) has the integrality property then variable splitting does exactly as well as linear programming. If the constraints placed into the objective by Lagrangian relaxation have the integrality property then variable splitting does as well as the natural Lagrangian relaxation.

In cases where the subproblems do not have the integrality property, extra constraints that are valid for the integer program but not for the linear program are often known. These valid inequalities can be written in terms of the original variables. This gives extra information about the original problem beyond the estimate of the objective function value.

It is still necessary to solve the relaxation of the problem that results from variable splitting. Any technique used for

Lagrangian relaxation can be employed. The most common method is subgradient optimization (Fisher (1985)). Unfortunately, this method is not guaranteed to find the optimal solution in any finite amount of time. Furthermore, the parameters required for this technique must be fine tuned for adequate performance. Despite these drawbacks, subgradient optimization has worked fairly well in practice. PDRC Report 86-10 suggests the ellipsoid algorithm as an alternative solution method. That report shows that the ellipsoid algorithm can be adapted to get an optimal solution in a polynomial amount of time. Furthermore, the method has no parameters to fine tune. While the algorithm is slower in finding the optimal solution on the sample problems solved, very good solutions were found very quickly. Since good solutions are enough for many purposes, the ellipsoid algorithm may be a practical method for these problems.

Many problems are amenable to variable splitting. The application to the TSP has already been mentioned. Nemhauser and Weber applied variable splitting to the set covering problem. This method can be specialized for the uncapacitated facility location problem. Integer programs with a small number of constraints can be divided into a small number of knapsack problems by variable splitting. Finally, many practical problems combine two or more well studied problems, like scheduling and material allocation. Splitting the variables in this case can permit the use of known heuristics rather than the alternative of creating a new heuristic.

In the next year or so, we want to address the following questions:

1) What other practical problems can variable splitting attack?

2) Under what conditions does variable splitting give the true objective function of the original optimization problem? Failing optimality, can it be proved that the results from variable splitting will not be too far from optimum?

3) How can the special structure of the relaxed constraints in a variable splitting problem be exploited? This would provide an alternative to subgradient optimization and the ellipsoid algorithm.

4) What interesting classes of valid inequalities can be created by variable splitting for well known problems? The constraints generated by Nemhauser and Weber's variable split for the set covering problem are not obviously a subset of known classes of valid inequalities, but that must be examined in more detail.

Bibliography

Baker, B., E.G. Coffman, and R.L. and Riest, "Orthogonal Packings in Two Dimensions" SIAM Journal on Computing, 9,4, 1980.

Fisher, M.L., "An Applications Oriented Guide to Lagrangian Relaxation," Interfaces 15, 2, 1985.

Gary, R. and D. Johnson, Computers and Intractability, Freeman Press, 1979.

Geoffrion, A.M. "Lagrangian Relaxation and Its Uses in Integer Programming," Math Programming Study 2, 1974.

Hadlock, F.O., "minimum Spanning Forests of Bounded trees," Proceedings of the Fifth Southeastern Conference on Graph Theory and Computing, 1974.

Leighton, F.T., "A Layout Strategy for VLSI Which is Provably Good," Proceedings of the 14th ACM Symposium on Theory of Computing, 1982.

Lipton, R. and E. Tarjan, "A Separator Theorem for Planar Graphs," SIAM Journal of Applied Math, Vol 36, No. 2, 1979.

E-24-648

# 1986 ANNUAL REPORT

E-24-648

Production and Distribution

Research Center

Submitted to

The Office of Naval Research

By


The School of Industrial and Systems Engineering

Georgia Institute of Technology

Atlanta, Georgia 30332

H. D. Ratliff

Principal Investigator

SSN: 424-64-4688

Telephone: (404) 894-2300

TABLE OF CONTENTS

# I. INTRODUCTION

The development of methodology to aid in the design and operation of systems for storing and moving material continues to provide some of the most important and challenging research issues faced by both military and civilian enterprises. Material movement systems are expensive to build and operate and have a tremendous impact on the activities which they support. They have become increasingly critical to U.S. military planning and operations due to the need to effectively support military activities throughout the world with decreased reliance on the assistance of other countries. The rapid evolution of technology related to material movement (e.g. computers and automated equipment) have also made research into design and operation of material movement systems critical to U.S. industry in trying to meet foreign competition.

For the past six years the research program supported by the Office of Naval Research in the Center for Production and Distribution Research (PDRC) at Georgia Tech has provided a focal point for research in developing new mathematical methodology to address problems associated with material movement and storage. Our research has focused on the following areas: (1) developing mathematical structures which enhance our understanding of the issues actually faced by practitioners in designing and operating distribution and logistics systems, (2) developing models and methodology which can be combined with the creative capabilities of human decision makers to facilitate the planning and design of distribution and logistics systems, (3) creating procedures and algorithms which, if not optimum, are predictable in the quality of the solutions generated for operational distribution and logistics problems, and (4) using mathematical optimization structures as bases for design of distribution and logistics systems which can be effectively operated and controlled.

## II. SUMMARY OF ACCOMPLISHMENTS

The ongoing basic research program within the PDRC continues to have a very significant impact on the role of mathematics in addressing problems related to material movement and storage. In addition to producing numerous reports (Appendix A), refereed publications (Appendix B), and invited presentations at national meetings (Appendix C) on research results, the PDRC continues to have an impressive array of visitors from both the military and private industry (visitors last year included Major General Archer Durham, Director of Deployment, Joint Deployment Agency, MacDill AFB; Major General Alan Salisbury, Commander of Information & Systems Engineering Command, Fort Belvoir; and Mr. Donald Peterson, Chairman of the Board of Ford Motor Co.).

The basic research in interactive optimization was instrumental in initiating the parallel effort which has been ongoing with the Joint Deployment Agency (JDA) for the past four years. This research has lead to the development of the MODES system for planning military deployments which is currently being tested at JDA. The basic research in two dimensional packing discussed in the next section has also lead to a new research effort which began this fall with the Military Airlift Command to develop mathematical methodology to aid in loading planes. Fundamental research results in vehicle routing which were developed in the PDRC have been implemented in routing systems in private industry including Coca-Cola and Ford Motor Co. These accomplishments are an indication of the value to both the military and private industry of the basic research program underway in the PDRC.

The central theme of our research involves the identification of spatial and combinatorial characteristics which allow reductions in the dimensionality of the problems. These reductions in dimensionality take a variety of forms. For order picking problems, the restricted travel within the warehouses frequently leads to efficient optimum solution techniques. For delivery problems, the space filling curve concept allows the problem to be mapped into a problem on a line. This cannot be solved optimally, but does lead to heuristics with provable bounds. For the interactive optimization approaches to layout and scheduling, optimization models are able to handle certain aspects of the problem, thus leaving the human designer with a much simpler problem to address. PDRC Reports 86-12 and 86-11 reflect our current research in order picking and interactive scheduling. PDRC Report 86-09 gives new results related to network aggregation. A PDRC report reflecting new results in interactive layout is currently being written.

During the past year we have also began to examine similar concepts in physical aggregation problems (e.g., product assembly, and consolidation/sorting of material for shipment), decentralized control of automatic guided vehicles, two dimensional packing problems, and partitioning of composite optimization problems. These problems are discussed in more detail below.

Product Assembly

Assembly of a product from its components is essentially an aggregation process which takes components and aggregates them. To complete the assembly in the minimum elapsed time requires that tasks be performed in parallel whenever possible. Since a decision on an assembly sequence requires information regarding feasible subassemblies, we require a compact representation of the feasible assembly region. One compact representation of the feasible subassembly region of a product, is obtained by constructing a tree graph. Every node in the graph corresponds to a component in the product. An edge in the graph implies an assembly process. Complete assembly of the product requires all assembly operations implied by the edges to be performed. If every connected component of the tree is considered to be a feasible subassembly of components, then all possible subassembled structures of the product are also connected components in the graph. The graph representation also implies that not more than two subassemblies may be involved in an assembly process. Furthermore, in the general case there may be different assembly times depending on the components being assembled. This can be represented as weights on the edges of the graph.

We have analyzed the problem of minimum time assembly of products whose assembly can be represented by a tree structure where all assembly operations are assumed to take unit time. When two components are assembled, the feasible region of the components not involved in the assembly and the subassembly formed are represented as follows. The edge corresponding to the assembly operation is "collapsed" and the components involved in the assembly operation are combined to form a super node. The super node is connected to those components in the rest of the graph that each of the components involved in the assembly were connected to. All feasible assembly structures in the new set of available components are represented by the new graph created. Furthermore, the new graph created is an 'aggregation' of the original graph structure.

When an assembly operation is being performed, components' not involved in the assembly process (some of the components may be subassembled components) may also be assembled in parallel operations. Thus if an operation were restricted to involve at most two components, the set of feasible assembly operations that can be done in parallel at any time could be represented by a 'matching' of edges in the corresponding graph. Since a matching identifies a set of edges that are disjoint (i.e. no two of them are incident to the same node), it also identifies sets of feasible parallel assembly operations.

Since each step in the assembly process aggregates components, performing only feasible assembly operations, the product (fully assembled) would be represented by a graph that is a single super node. The minimum time assembly problem can therefore be described as finding the minimum number of steps, each step involving a matching on the aggregated graph from the earlier step, required to collapse the graph into a single node. We observe that performing a maximum matching of assembly operations at each step may not yield the minimum number of steps.

The assembly operations and feasible matchings can be described in terms of the original graph itself as edge labeling operations that satisfy the property that between two edges of the same label i, there is at least one edge with a label $j > i$. This problem represents a constrained version of the edge coloring problem on a tree graph. Since edge coloring on a tree can be solved by a linear time algorithm, an interesting research question is whether the constrained version of the problem is solvable quickly.

The same problem can also be described as that of accepting a tree graph G as input and creating a binary tree T of minimum height such that the vertices of G are the leaf nodes of T and the edges of G are the internal nodes of T. In this framework, the problem is related to the edge separator tree of a tree, described by Leighton (1982) in the context of circuit layout.

Some of the results obtained regarding this problem are as follows.

(1) For special tree structures such as complete binary trees, chains of stars, stars, chains, star of chains etc. optimal algorithms have been obtained.

(2) For the weighted chain structure, a polynomial algorithm is developed that provides the minimum time assembly of the

components.

(3) Given a set of optimally labeled subtrees all connected to a single node, we can label the new graph (consisting of the original subtrees and the edges linking it to the additional node) by using no more than one label over the optimal i.e.

$HEUR( i + T_1 + \cdots + T_k) \leq OPT( i + T_1 + \cdots + T_k) + 1$

(4) For any arbitrary tree structure, we can use the node separator tree of Lipton and Tarjan (1979), which is guaranteed to be of height <= O(log n), to develop an algorithm with a guaranteed constant worst case bound.

Consolidation and Sorting in Distribution Systems

In material movement systems where packages are picked up at multiple origins and delivered to multiple destinations, aggregation of packages in regions for re-distribution is a common strategy used. Decisions have to be made regarding the regions formed, re-distribution strategies, sorting mechanisms etc. Typically binding constraints are that the volume handled by each of the regions formed be reasonable, and that the time to deliver a package from any one point to another be guaranteed to be less than a certain input value. Other constraints include availability of resources i.e. number of resources, capacity etc. In general we have a complete graph structure on the nodes (being points of origin of material), and also information regarding the flow from point i to point j.

Consider a tree structure linking the points where packages are to be delivered. Also, suppose there is a parameter associated with each edge which can be used to separate the node set into two connected subtrees. This implies that each destination on n nodes has associated with it an n-1 size vector of 0 or 1 with respect to the parameters on each of the n-1 edges. Now suppose all the material to be delivered is accumulated at a point. The minimum height binary tree discussed in the earlier section would also provide the smallest number of sorting steps required to separate the packages into individual entities, where a sorting step accepts a group of packages and separates them into two groups based on the value of a parameter being 0 or 1.

Consider a sorting mechanism which has a complete binary tree structure. A parameter can be set on each of the internal nodes. The whole set of packages is introduced at the root node and get separated out into groups based on the parameter settings at the internal comparator nodes. Now, if the minimum height of the binary tree required to separate the entities is lower than the sorter height, then the sorting can be performed in using one sorter. However, if we are given a fixed height binary sorter, we need to decide on the location and inputs to various sorters, while minimizing the number of such sorters used. Even if we have a minimum height binary tree, we may not have a layout of the nodes that minimizes the number of sorters used. However, for a given tree layout, we provide a linear time algorithm that produces the minimum number of sorters necessary and their location.

Consider the basic problem of locating sorting stations that service a group of nodes that are connected. The sorting station essentially operates in two phases. In the first phase, the packages originating from and destined for nodes in a subtree are sorted down to their individual levels. However, packages headed from this subtree region to other subtree regions are represented as aggregate groups, one for each of the regions. In the next phase, all these aggregate groups are re-distributed and all packages addressed to nodes in a region are accumulated. Then, this aggregate set is sorted down to its individual entities. Therefore, the sorting station capacity would influence the total flow exchanged within a region and also entering a region from other regions. We show that even on a tree structure where flows are exchanged only between adjacent nodes, the problem is NP-complete Garey and Johnson (1979), by reduction from the two partition problem.

The next problem is that of re-distribution given a constraint on the total flow into each subtree. This implies that for each node we compute the total flow into it from all other nodes and that all the sorting into detailed entities is done after all the re-distribution is completed. However, the time it takes for every package to get from one point to another is constrained. If this constraint were relaxed, then Hadlock's (1974) algorithm would provide a partition of the subtree into a minimum number of stations. Since re-distribution of the packages can be considered to be proportional to the distances on the cut edges between subtrees (the tree structure implies that there is at most one such edge between any pair of subtrees), the problem is to partition the tree into subtrees of capacity <= W while the sum of the flows on the cut edges is <= T. This problem is also NP-complete by reduction from the knapsack problem.

Therefore both the load balancing problem for sorting stations and the time constraint on the package delivery make the problem NP-complete. Hence, we consider cases where only one or the other constraint applies. Consider the time constraint on the flows between points. The flow process consists of accumulating packages within each subtree region, then re-distributing packages across regions and finally sorting them down to their individual levels. If a unit of collection resource were available at each subtree and at the higher inter-regional level, the problem becomes that of dividing a tree into regions so that the sum of the edge weights in each region is constrained <= W and the sum of the cut edge weights is constrained to be <= T. This problem again is NP-complete by two-partition.

An alternative system would be for packages to move independently to the center of their corresponding subtree regions (radius of regions being constrained) and for the sum of the cut edge weights to be <= T. The complexity of this problem is being examined.

Decentralized Control of Automatic Guided Vehicles (AGV's).

The control of AGV's is proving to be a very good and practical arena in which to test our ideas on decentralized, locally autonomous optimization. The automated factory or warehouse which uses AGV's can be seen as a hierarchical, real time environment in which decisions must be made locally, for speed and reliability of the system. We are studying how to decentralize optimization/control and then how to integrate the responses of the subsytems.

We have been helped in this by cooperation from Litton Industries, a major manufacturer of AGV's. With them we have developed a classification scheme for AGV systems and control strategies, and for each system we are establishing performance bounds for different control strategies. Our results so far indicate that even simple decentralized control heuristics can be guaranteed to work surprisingly well.

We have studied the "Busy Heuristic" (discussed in the previous proposal), and have additionally determined that, while it is guaranteed to perform exceptionally well under the criterion of "latest delivery time", it is capable of poor performance for average throughput. (This is to be expected since the Busy Heuristic is the most myopic practical control strategy.) To understand how to ensure good throughput, we have analyzed the "Soonest Delivery Heuristic", which, when there is space available aboard the AGV, next picks up the item which is closest to its destination. Notice that this requires more look-ahead than does the Busy Heuristic, but it is nevertheless sufficiently decentralized to allow different AGV's to operate independently of each other; they need query only the load/unload stations, and not the other vehicles.

We have proven that, under the Soonest Delivery Heuristic the following is true.

1. The time of the last delivery of a tote will be no more than twice the optimum time.

2. The AVERAGE time of delivery of all totes will be no more than twice the optimum.

In this analysis, "optimum" is unrealistically small. It is calculated for the ideal case in which the AGV can randomly access any tote at a load/unload station (rather than observe queue order), and totes can be temporarily deposited at intermediate stations, to be delivered later.

Now we are studying "priority-based" heuristics, wherein totes of high priority are given first consideration for delivery.

We are also studying the effects of network topology on the performance of the scheduling rules. Here our results are facilitated by the fact that most real AGV networks are "series-parallel". Since such networks have simple recursive structure, we can assign part of the routing control functions to components of the network. Then distinguished nodes of the network can work with the simple AGV control to achieve high quality routing, without centralized computation. For example, a portion of a typical AGV network includes multiple lanes for pickup/deliveries. In this component of the AGV network, install a linked pair of autonomous controllers at entry point A/exit point B. We assign these controllers responsibility for only and exactly the paths between A and B. Now these controllers can work with, say, the Busy Heuristic to route the AGV appropriately: at A the AGV receives instructions as to which lane to travel; through the lane, the AGV uses its local "intelligence" to determine pickups and deliveries; at B the AGV reports what pickups were made; B relays this information back to A, so that A can appropriately route the next AGV. We have preliminary results on the performance of such configurations that show that worst-case performance can be bounded by small factor that depends on the complexity of the structure of the AGV network (which in practice is small).

This research will be presented at an invited presentation to the ORSA/TIMS Joint National Meeting in Miami, 1986. It is will be reported on more thoroughly in the doctoral dissertation of Wang Lim (expected date of graduation: summer 1987).

## Two-dimensional Packing

Spatial packing problems are a good test area in which to explore issues of autonomous optimization because spatial problems can be decomposed simply and naturally. Typically, packing in one region affects only slightly the packing in more distant regions. "Shelf heuristics" attempt to exploit this directly, by partitioning the region to be packed into "shelves" which then can be packed independently and in parallel. A centralized intelligence is necessary only to assign each item to be packed to the appropriate shelf.

A further advantage of shelf heuristics is that they result in subproblem simplification: each shelf can be packed well by simple procedures since the allocator has screened the items sent there to be packed. For example, each shelf can be packed as a simple 1-dimensional packing problem, and still yield good results. In addition, the simplicity of the subproblem enables the packing to be computed "on-line", since the entire problem need not be known in advance. The partitioning ensures that there will be little significant interaction among the items in the shelves (for large problems).

Thus, when the problem is decomposed appropriately, items allocated appropriately, and subproblems solved appropriately, the problem can be solved well at the highest level. was first studied by B. Baker, E. G. Coffman and R. L. Rivest (1980). They formalized the problem as follows: Given a bin with fixed width and unbounded height, and a list of rectangles ("boxes"), pack the boxes into the bin so that no two boxes overlap and so that the height to which the bin is filled is as small as possible. We assume that the boxes are "oriented", each having a specified side that must be parallel to the bottom of the bin. 2-dimensional bin-packing has widespread applications, including stock cutting, warehouse storage, scheduling with resource constraints, etc. Because of the intractability of packing, research has concentrated on analyses of heuristics. The heuristics can be classified into two main types:

1. Off-line algorithms are allowed to defer the packing of boxes and rearrange boxes that have already been packed. This corresponds to the algorithm having complete knowledge in advance of the problem, so that the problem can be solved completely before the solution is implemented.

2. On-line algorithms must pack boxes in the same order as they are listed. This is the case when the algorithm must solve the problem and implement the solution even as the problem is being revealed. For example, imagine the boxes to be packed arriving over time. An on-line algorithm must pack the boxes as they arrive, and without knowing anything about future arrivals.

On-line heuristics are less accurate in general than off-line heuristics since they must work with less information. Also they are constrained to pack boxes according to the sequence in which they appear on the list.

As is generally the case, worst-case bounds are more easily available than expected case bounds. Worst-case bounds are either

1. asymptotic, i.e. of the form Heuristic $<$ a * OPT + b, or
2. absolute, i.e. of the form Heuristic $<$ a * OPT.

Absolute bounds are generally not as good as asymptotic bounds, since the usual heuristics can typically be fooled badly on very small, skewed problems. This is because most heuristics do some sort of partitioning, and the problems must be "large enough" so that the partitions are fully used.

We generalize the computer science model of bin-packing to this: Given a set of storage areas of specified shapes and dimensions, pack a list of boxes into the storage areas so as to waste as little space as possible. Our results are for a special type of on-line heuristic known as a "shelf" heuristic. The idea is to pack boxes onto "shelves", which are constructed as needed. A central issue is to decide what size shelves to use. After it is constructed, each shelf can be treated as a bin of one less dimension. (For example, in 2-dimensional bin-packing a shelf is simply a 1-dimensional bin, where the single dimension is the width of the shelf). This enables us to reduce higher dimensional packing problems to simpler, lower dimensional packing problems - however, at some cost in accuracy of the solution. Our results suggest that in practice this trade-off can frequently be worth making. Our results are strong in that we compare heuristic performance with the unrealistically good solution possible when boxes are completely deformable, and so can be packed with absolutely no wasted space (as if we were packing water rather than boxes). This is a lower bound even on the very best possible when the future is completely known and we are willing to devote whatever computing power necessary to take advantage of that knowledge. Our results so far include:

1. An analysis of how the set of shelf sizes affects the performance of Baker and Schwartz's heuristic. A result of this is a better set of shelf sizes that lead to improved bounds. Furthermore we show lower bounds on the worst-case performance of any shelf heuristic, independent of the specific shelf sizes.

2. Average-case performance analysis for Baker and Schwartz's shelf heuristic. The result shows how the shelf sizes can be chosen to achieve good worst-case performance but at the cost of diminished expected performance, or to achieve good expected performance at the cost of diminished worst-case performance. Thus worst-case and expected performance can be balanced in advance by the user.

3. Analysis of the shelf heuristic in the special cases of when

a. the storage area consists of a set of square bins (rather than a single open-ended bin).

b. when the largest and smallest dimensions of the boxes are known a priori. (Considerable better bounds are possible when the dimensions of the smallest box are not "too" much smaller than those of the largest box to be packed. This would be the case in a warehouse.)

4. How to determine the best shelf sizes when only a fixed, finite number of shelf sizes are allowed and a probability distribution is known for the dimensions of arriving boxes. Results for when the storage area is either

a. a single open-ended bin, or

b. finite set of squares.

5. How to determine the optimal shelf sizes for the "dynamic bin-packing problem", in which boxes enter the system, are packed and remain in storage for some time, and then leave the system in the following cases:

a. we know a priori probabilistic information about arrivals and departures of boxes;

b. we have no a priori knowledge of arrivals and departures of boxes.

6. Worst-case analysis of a "shelf and floor" heuristic for 3- dimensional packing when the storage space is either

a. a single open-ended bin, or

b. a set of cubes.

In addition to the above results, we have a number of partial results that we expect to resolve. We are also broadening the problem to incorporate physical constraints, such as requiring "stable" packings (so boxes will not tip over), even distribution of weight, etc.

This work will be reported on in the doctoral dissertation of Zhang Jixian (expected date of graduation: August 1987).

Voting Procedures

We have also begun studying how to combine subproblem solutions so that their quality is preserved and transmitted to higher levels in the hierarchy of problem components. In highly decomposable systems, such as the packing problems above, integrating subproblem solutions is trivial. For less decomposable systems, the issues are considerable more complex. In such systems, we can imagine each autonomous optimizer to produce a "preference ordering" of alternative solutions. Then the higher level controller is like a "voting procedure", which intelligently integrate these subproblem solutions to produce a group decision. What sort of "voting procedures" should be used, and when can they be guaranteed to work acceptably?

In principle such procedures are susceptible to the anomalous behavior of known to apply to social voting systems; but our procedures need not be constrained to satisfy criteria of "fairness" or "social rationality". Given this, extra leeway, can we design functional procedures to integrate the recommendations of the autonomous optimizers?

We have identified some paradigms in which this is possible. For example, the Stable Matching problem asks to pair 2n nodes so that no 2 nodes prefer each other to their current partners. In a sense each node is an autonomous decision maker who must accommodate the preferences of all others. In general there is no solution to this problem. However, given a structure on the preferences which plausibly models the psychology of a human formulator of criteria, a stable matching always exists, is unique, and can be constructed quickly under the auspices of a very weakly centralized algorithm which simply coordinates the self interest of the individual nodes. Furthermore, since the algorithm relies so much on the self interest of the nodes, it is highly parallelizable. (This has been reported on in "Stable matching with preferences derived from a psychological model" by John J. Bartholdi, III and Michael A. Trick, to appear in Operations Research Letters.) We expect more of this type of result, which give non-trivial ways of integrating the preferences of subsystems, but while preserving a high degree of decentralization.

We remark parenthetically that this line of thought has led to an unexpected result that is of considerable interest for social decision-making: we have proved that a clearly defined, historical voting scheme has the property that it can be "hard" to determine the winner! (Specifically, to determine the score of any candidate is NP-complete, and to determine the winner is $D^P$-equivalent.) This result is apparently the first to introduce computational complexity into social decision-making; and, moreover, is interesting for historians of voting procedures. This will be reported on in "The Complexity of Voting" by John J. Bartholdi, III, Craig A. Tovey, and Michael A. Trick.

Another approach to integrating subproblem solutions is to allow the system to "evolve" a procedure. For example, one might simulate the system, measure performance, and feed this back to the local decision makers, who are provided with the ability to mutate in response to the feedback. Given the right evolutionary pressure, the system will presumably evolve toward efficient performance. We have established just this sort of behavior for a system based on iterated plays of a formalized game, reported on in "More on the evolution of cooperation" by John J. Bartholdi, III, C. Allen Butler, and Michael A. Trick, Journal of Conflict Resolution, Vol. 30 No. 1, March 1986, pp. 129-140.

Solving Optimization Problems by Variable Splitting

Many difficult optimization problems contain large subproblems that, if solved independently, are tractable. The difficulty arises in the relationships among the subproblems and, possibly, difficult extra constraints. The deployment planning problem is such a problem. It consists of a transportation configuration problem and a movement requirement assignment problem. Each of these problems considered alone (i.e. with the variables in the other problem fixed) is a minimum cost network flow problem. Taken together, the problem is a linear programming problem, not a network flow problem. Another example is the Travelling Salesman Problem (TSP). Here the goal is to find a minimum length route that visits each of a set of cities exactly once. One subproblem of the TSP is to find a set of links between cities (edges) so that each city is adjacent to two edges. Another subproblem is to find a set of edges that joins all of the cities. Each subproblem is easy to solve alone (the first by matching techniques, the second by spanning tree methods), but together the TSP is notoriously difficult to solve.

In most cases where there are large, easily solved subproblems, techniques are available to exploit the structure. In some cases (like the deployment problem) these techniques are more effective methods for finding the optimal solution. In other cases (like the TSP) exploiting the structure of the subproblem is not guaranteed to provide the optimal solution to the original problem. However, good estimates of the optimal cost are given. These estimates can be incorporated in other techniques (like Branch and Bound), enhancing the search for the optimal solution.

One general method for attacking these problems is Lagrangian relaxation (Geoffrion (1974)). Lagrangian relaxation can be applied where there is a large, obvious, portion with exploitable structure and a small number of complicating constraints. In this case, the complicating constraints are multiplied by some cost, representing the cost of removing that constraint, and placed into the objective. In this case, some of the cost is allocated to the subproblem and the rest is allocated to the relaxed constraint, in the form of a constant.

Lagrangian relaxation has proved effective in a wide variety of practical problems (Fisher (1985)). One fruitful area of research has been methods to expand the applicability of Lagrangian relaxation. For instance, Lagrangian relaxation cannot be applied directly to the TSP formulation given above because both subproblems have a enormous number of constraints, precluding placing either set into the objective.

Variable splitting is a promising, general technique for bringing out exploitable structure when it is hidden in problems. In variable splitting, each variable is replaced by two or more new variables. Each constraint involving the old variable receives one of the new variables instead. If all of the new variables are forced to take on the same value, then the new problem is exactly the same as the old problem. If variable splitting is done cleverly then a problem can be created that is amenable to Lagrangian relaxation or some other technique to exploit special structure.

For instance, the TSP can be written as the following integer program:

$$\begin{array}{ll} \text{Minimize} & c\,x \\ \text{Subject to} & \\ & A_1\,x \le b_1 \\ & A_2\,x \le b_2 \\ & \quad x \ge 0,\ \text{integer}, \end{array}$$

where the $A_1$ constraints represent the requirement that two edges hit each city and the $A_2$ constraints force the tour to be connected. Each variable $x_i$ can be replaced with two variables $y_i$ and $z_i$. Every occurrence of $x_i$ in $A_1$ is replaced with $y_i$ while those in $A_2$ are replaced with $z_i$. Constraints forcing $y_i$ to equal $z_i$ are added resulting in the problem:

$$\begin{array}{llll} \text{Minimize} & c\,y & & \\ \text{Subject to} & & & \\ & A_1\,y \le b_1 & & \\ & & A_2\,z \le b_2 & \\ & y & -\;z = 0 & \\ & y, & \;\;z \ge 0,\ \text{integer}. & \end{array}$$

Now the $y-z = 0$ constraints can be relaxed by Lagrangian relaxation. Note that the number of constraints relaxed depends only on the number of variables, not the number of constraints in

$A^1$ or $A_2$. Also, the problems in y and z are completely separate, so the structure of $A_1$ and $A_2$ can be exploited.

The problem created by splitting the variables and relaxing the constraints forcing the new variables to be equal is called the variable splitting problem.

Some questions that we have addressed in the past year are:

1) What is the relationship between variable splitting and other relaxations, such as linear programming and Lagrangian relaxation on the original problem?

2) Does variable splitting give any extra information about the original optimization problem beyond the estimate of the optimal objective function value?

3) How can the problem that results from performing Lagrangian relaxation after variable splitting be solved?

4) What sort of problems are amenable to variable splitting?

The key to comparing variable splitting with other relaxations is to see relaxations as optimizing over an expanded set of feasible points. The original optimization problem is concerned with optimizing over the smallest feasible region. The linear programming relaxation optimizes over a region at least as large. We can show that no matter how the variables are split, the region optimized over is no larger than the linear programming region. This implies that variable splitting is at least as good as linear programming. In cases where the problem has a natural Lagrangian Relaxation, creating one set of variables for the specially structured subproblem and another for the relaxed constraints is a natural variable split. In this case, the region for the variable split problem is contained in the region for the natural Lagrangian relaxation, so variable splitting is at least as good as Lagrangian relaxation.

This sort of analysis also shows when the relaxations will give the same results: when the regions are the same. A key concept is the integrality property. A problem has the integrality property when dropping the integrality restrictions does not affect the optimization property. Some examples of problems with integrality property are network flow problems and bipartite matching problems. If the problem created by variable splitting (relaxing the constraints that force the new variable to equal each other) has the integrality property then variable splitting does exactly as well as linear programming. If the constraints placed into the objective by Lagrangian relaxation have the integrality property then variable splitting does as well as the natural Lagrangian relaxation.

In cases where the subproblems do not have the integrality property, extra constraints that are valid for the integer program but not for the linear program are often known. These valid inequalities can be written in terms of the original variables. This gives extra information about the original problem beyond the estimate of the objective function value.

It is still necessary to solve the relaxation of the problem that results from variable splitting. Any technique used for Lagrangian relaxation can be employed. The most common method is subgradient optimization (Fisher (1985)). Unfortunately, this method is not guaranteed to find the optimal solution in any finite amount of time. Furthermore, the parameters required for this technique must be fine tuned for adequate performance. Despite these drawbacks, subgradient optimization has worked fairly well in practice. PDRC Report 86-10 suggests the ellipsoid algorithm as an alternative solution method. That report shows that the ellipsoid algorithm can be adapted to get an optimal solution in a polynomial amount of time. Furthermore, the method has no parameters to fine tune. While the algorithm is slower in finding the optimal solution on the sample problems solved, very good solutions were found very quickly. Since good solutions are enough for many purposes, the ellipsoid algorithm may be a practical method for these problems.

Many problems are amenable to variable splitting. The application to the TSP has already been mentioned. Nemhauser and Weber applied variable splitting to the set covering problem. This method can be specialized for the uncapacitated facility location problem. Integer programs with a small number of constraints can be divided into a small number of knapsack problems by variable splitting. Finally, many practical problems combine two or more well studied problems, like scheduling and material allocation. Splitting the variables in this case can permit the use of known heuristics rather than the alternative of creating a new heuristic.

In the next year or so, we want to address the following questions:

1) What other practical problems can variable splitting attack?

2) Under what conditions does variable splitting give the true objective function of the original optimization problem? Failing optimality, can it be proved that the results from variable splitting will not be too far from optimum?

3) How can the special structure of the relaxed constraints in a variable splitting problem be exploited? This would provide an alternative to subgradient optimization and the ellipsoid algorithm.

4) What interesting classes of valid inequalities can be created by variable splitting for well known problems? The constraints generated by Nemhauser and Weber's variable split for the set covering problem are not obviously a subset of known classes of valid inequalities, but that must be examined in more detail.

## Bibliography

Baker, B., E.G. Coffman, and R.L. and Riest, "Orthogonal Packings in Two Dimensions" SIAM Journal on Computing, 9,4, 1980.

Fisher, M.L., "An Applications Oriented Guide to Lagrangian Relaxation," Interfaces 15, 2, 1985.

Gary, R. and D. Johnson, Computers and Intractability, Freeman Press, 1979.

Geoffrion, A.M. "Lagrangian Relaxation and Its Uses in Integer Programming," Math Programming Study 2, 1974.

Hadlock, F.O., "minimum Spanning Forests of Bounded trees," Proceedings of the Fifth Southeastern Conference on Graph Theory _and Computing_, 1974.

Leighton, F.T., "A Layout Strategy for VLSI Which is Provably Good," _Proceedings of the 14th ACM Symposium on Theory of Computing_, 1982.

Lipton, R. and E. Tarjan, "A Separator Theorem for Planar Graphs," _SIAM Journal of Applied Math_, Vol 36, No. 2, 1979.

SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING
PRODUCTION AND DISTRIBUTION RESEARCH CENTER
Atlanta, Georgia 30332 (404) 894-2339

July 22, 1988

Dr. Neal Glassman
US Office of Naval Research
800 N Quincy Street
Arlington, VA 22217

Dear Neal:

Enclosed is our annual report together with reprints of publications since the last report. I am working on the new proposal and will have it to you within a couple of months. If you need anything else, please let me know.

Sincerely yours,

H. Donald Ratliff
Regents Professor

HDR/amr

PROGRESS REPORT
**ONR CONTRACT N00014-86-K-0173**
H. Donald Ratliff P.I.
July 20, 1988

Research under the ONR sponsored Production and Distribution research center has continued to provide creative new mathematical methodology for solving problems associated with production and distribution. Refereed journal papers published in 1987-88 which describe research under the project are listed in section I and the reprints are attached. Papers accepted for publication but not yet published are listed in section II. Papers currently in the review process are listed in section III. PDRC technical reports are listed in section IV. Invited presentations related to the various papers are listed in section V. Several paper deserve comment since they seem to be of unusual significance.

There are three papers [I.2],[I.3]and [1.5] which address fundamental mathematical problems associated with the picking operation in a warehouse. In [I.2] a polynomial time algorithm is developed for determining the optimum pick sequence in an aisle. The resulting picking tour is shown to be as much as 30% better than the heuristic most commonly used. In [I.3] an algorithm is developed to determine the optimum stop locations for a picking vehicle for manual picking operations. Both of these algorithms can be efficiently run on a personal computer. In [I.5] bin numbering schemes based on "space filling" curves are developed which allow very good picking tours to be generated based on a simple sort. We believe that these three concepts will allow very significant savings in the primarily manual picking systems found in the vast majority of warehouses.

There are two other papers [II.9] and [III.3] which address storage issues in a warehouse. The concept developed in [II.9] seems particularly important. The concept involves determining storage locations based on the length of time that the item will be in the warehouse. This is fundamentally different from any storage concept developed previously. We show that at least from a mathematical prospective, this concept dominates all previous storage methodologies. We believe that this will ultimately make substantial improvements in automated warehouses where items handled in pallet loads.

The results in [I.1], [1.6] and [1.7] show how to incorporate sophisticated mathematical optimization methodology (i.e., matching, cut trees, and location theory) into the facility design process. This is a very important area where the actual use of mathematics in design has been very limited up to now. We believe that these results provide a major step in advancing mathematics in design.

In [III.4] and [III.6] we have some initial success in developing graph theoretic models to address design of distribution systems and assembly systems. The theoretical results also appear to have important implications related to "design for assembly." This is a particularly exciting area for additional work. There is almost no previous work with a solid mathematical basis and the results appear to be significant.

There are three papers [II.1],[II.2], and [III.7] which introduce complexity theory to social choice and prove several novel theorems that complement the famous Arrow theorem on the impossibility of fair voting schemes. In [II.1] it is shown that two historical voting schemes have unfortunate property that it is NP-hard to tell whether any particular candidate has won the election. (One of these voting schemes is due to Lewis Carroll, who seriously suggested it for Oxford University; we think he would have liked this result.) We also prove an "impracticality" theorem, which is a computational analogue of Arrow's theorem. Our theorem says any voting rule that does enough work to meet certain minimal conditions of fairness must do so much work that it is impractical; that is, it is NP-hard to determine who won the election! In [II.2] we discuss another famous impossibility theorem due to Gibbard and Satterthwaite; it says that any voting scheme that meets certain minimal conditions of fairness must be susceptible to manipulation by strategic voting. We give an algorithm that will manipulate most commonly-used voting schemes in polynomial-time. More interestingly, we show a real voting rule that is in principle susceptible to manipulation by strategic voting, but is computationally resistant; that is, it is NP-hard to determine how to manipulate! This suggests that computational complexity can protect the integrity of social choice. Even where abuse is possible, it might not be practical on computational grounds. (This is similar to current ideas in cryptography, where an ideal cryptographic scheme is easy to use but computationally hard to decrypt.) In [III.7] we study the computational complexity of several other forms of manipulating an election, and show how different voting schemes vary in their susceptibility. The main result is that, while Arrow's theorem says that four rationality criteria are inconsistent, for some voting schemes it is NP-hard to recognize inconsistencies! We expect that these three papers will have a considerable impact on social choice theory.

I. REFEREED PUBLICATIONS (Reprints Attached)

(1) "Matching Based Interactive Facility Layout," IEE Transactions, IIE Transactions, Vol. 19, No. 3, September 1987.

(2) "Order Picking in An Aisle," IIE Transactions, Vol. 20, No. 1, March 1988.

(3) "An Efficient Algorithm to Cluster Order Picking Items in a Wide Aisle," Engineering Costs and Production Economics, Vol 13, 1988.

(4) "Sequencing Picking Operations for a Man-Aboard Order Picking System, Material Flow, Vol. 4, No. 4, March 1988.

(5) "Design of efficient bin-numbering schemes for warehouses", Material Flow, Vol. 4, No. 4, March 1988.

(6) "A simple and efficient algorithm to compute tail probabilities from transforms", Operations Research, Vol 36, No. 1, 1988.

(7) "A Decomposition Procedure for Convex Quadratic Programs," Naval Research Logistics Quarterly, Vol. 35, 1988.

(8) "Heuristics based on spacefilling curves for combinatorial problems in Euclidean space", to appear in Management Science, Vol.34, March 1988.

II. PAPERS ACCEPTED FOR PUBLICATION

(1) "Voting schemes for which it can be difficult to tell who won the election", to appear in Social Choice and Welfare.

(2) "The computational difficulty of manipulating an election", to appear in Social Choice and Welfare.

(3) "Decentralized control of automated guided vehicles on a simple loop", to appear in IIE Transactions.

(4) "Spacefilling curves and the travelling salesman problem", to appear in J. Assoc Comput Mach.

(5) "Expected performance of shelf heuristics for 2-dimensional packing", to appear in OR Letters.

(6) "Optimizing the Location of Input/Output Stations within Facilities Layout," to appear in Material Flow.

(7) "On a Node Ranking Problem of Trees and Graphs," to appear in Information Processing Letters.

(8) "Utilizing Cut Trees as Design Skeletons for Facility Layout," to appear in IIE Transactions.

(9) "Shared Storage Policies Based on Duration of Stay," to appear in Management Science.

III. PAPERS IN REFEREEING

(1) "Interactive Optimization Methodology for Fleet Scheduling," NLRQ.

(2) "Hierarchical Solution of Network Flow Problems" Networks.

(3) "Determining Optimal Lane Depths for Single and Multiple Products in Block Stacking Operations," IIE Transactions.

(4) "Location Issues in Guaranteed Time Distribution Systems", Management Science.

(5) "On an Edge Ranking Problem of Trees and Graphs," Discrete Applied Mathematics.

(6) "Parallel Assembly of Modular Products - An Analysis," European Journal of Operations Research.

(7) "How hard is it to cheat in an election?", Econometrica.

## IV. TECHNICAL REPORTS

| | |
|---|---|
| 87-01 | Solving Large Generalized Networks, W. Nulty, M. Trick |
| 87-02 | Utilizing Cut Trees as Design Skeletons for Facility Layout, B. Montreuil, H. D. Ratliff |
| 87-03 | On a Node Ranking Problem of Trees and Graphs, A. V. Iyer, H. Donald Ratliff, G. Vijayan |
| 87-04 | Optimal Lane Depths for Single and Multiple Products in Block Stacking Storage Systems, M. Goetschalckx, H. D. Ratliff |
| 87-05 | Optimizing the Locations of Input/Output Stations Within Facilities Layout, Benoit Montreuil, H. Donald Ratliff |
| 87-06 | An Averaging Algorithm for Modes, Ananth V. Iyer, John J. Jarvis, H. Donald Ratliff |
| 87-07 | Hierarchical Approaches to Solution of Modes, Ananth V. Iyer, John J. Jarvis, H. Donald Ratliff |
| 87-08 | Location Issues in Guaranteed Time Distribution Systems, Ananth V. Iyer, H. Donald Ratliff |
| 88-01 | Conditions for Finite Convergence of Algorithms for Nonlinear Programs and Variational Inequalities<br>Faiz A. Al-Khayyal, Jerzy Kyparisis |
| 88-02 | Using Feasibility Cuts to Accelerate the Convergence of MODES<br>Moshe Eben-Chaime, John J. Jarvis, and H. Donald Ratliff |
| 88-03 | Plane Loading Algorithms for Military Airlift Command<br>J. Jarvis, H. D. Ratliff, C. Hane and B. Stutzman |
| 88-04 | Global Optimization of Concave Functions Subject to Separable Quadratic Constraints and of All-Quadratic Separable Problems<br>Faiz Al-Khayyal, Reiner Horst, and Panos M. Pardalos |
| 88-05 | Ship Scheduling by Sequential Solution of Shortest Paths<br>John J. Jarvis, Kevin L. McCroan, and H. Donald Ratliff |

## V. INVITED PRESENTATIONS

"Implementation Issues in Microcomputer Vehicle Routing," TIMS/ORSA Joint National Meeting, New Orleans, May 1987.

"Graph Theory Based Assembly Planning," TIMS/ORSA Joint National Meeting, New Orleans, May 1987.

"Location Issues in Guaranteed Time Distribution Systems," ORSA/TIMS Joint National Meeting, St. Louis, October 1988.

"Optimization Issues in Deployment Planning," EURO IX, TIMS XXVII, AFCET Joint International Meeting, Paris-France, July 1988.

Table Ronde sur l'Analyse de l'Aggregation des Preferences et des Choix en l'Honneur de Marquis de Condorcet, Centres International des Rencontres Mathematiques, Marseille-Luminy; "The computational difficulty of manipulating an election", April 1988.

Yale University, Graduate Seminar of the School of Organization and Management; "The computational complexity of social choice", November 1987.

The University of Florida, Graduate Seminar of the Department of Industrial and Systems Engineering: "The computational complexity of social choice", October 1987.

Massachusetts Institute of Technology, Graduate Seminar in Operations Research; "The computational complexity of social choice methods", April 1987.

The University of Arizona, Graduate Seminar in Decision Sciences; "The computational complexity of social choice methods", April 1987.

## VI. OTHER

Dr. H. Donald Ratliff is serving as Editor in Chief of the Journal of Operations Research.

Dr. John J. Bartholdi is serving as Departmental Editor for New or Non-traditional applications for the Journal of Operations Research and as Departmental Editor for Planning, Scheduling, and Control for the IIE Transactions.

FINAL REPORT
**ONR CONTRACT N00014-86-K-0173**
H. Donald Ratliff P.I.
June 30, 1989

Research under the ONR sponsored Production and Distribution research center has continued to provide creative new mathematical methodology for solving problems associated with production and distribution. Refereed journal papers which describe research under the project are listed in section I. Papers accepted for publication but not yet published are listed in section II. Papers currently in the review process are listed in section III. PDRC technical reports are listed in section IV. Invited presentations related to the various papers are listed in section V. Several paper deserve comment since they seem to be of unusual significance.

There are three papers [I.2],[I.3]and [1.5] which address fundamental mathematical problems associated with the picking operation in a warehouse. In [I.2] a polynomial time algorithm is developed for determining the optimum pick sequence in an aisle. The resulting picking tour is shown to be as much as 30% better than the heuristic most commonly used. In [I.3] an algorithm is developed to determine the optimum stop locations for a picking vehicle for manual picking operations. Both of these algorithms can be efficiently run on a personal computer. In [I.5] bin numbering schemes based on "space filling" curves are developed which allow very good picking tours to be generated based on a simple sort. We believe that these three concepts will allow very significant savings in the primarily manual picking systems found in the vast majority of warehouses.

There are two other papers [II.9] and [III.3] which address storage issues in a warehouse. The concept developed in [II.9] seems particularly important. The concept involves determining storage locations based on the length of time that the item will be in the warehouse. This is fundamentally different from any storage concept developed previously. We show that at least from a mathematical prospective, this concept dominates all previous storage methodologies. We believe that this will ultimately make substantial improvements in automated warehouses where items handled in pallet loads.

The results in [I.1], [1.6] and [1.7] show how to incorporate sophisticated mathematical optimization methodology (i.e., matching, cut trees, and location theory) into the facility design process. This is a very important area where the actual use of mathematics in design has been very limited up to now. We believe that these results provide a major step in advancing mathematics in design.

In [III.4] and [III.6] we have some initial success in developing graph theoretic models to address design of distribution systems and assembly systems. The theoretical results also appear to have important implications related to "design for assembly." This is a particularly exciting area for additional work. There is almost no previous work with a solid mathematical basis and the results appear to be significant.

There are three papers [II.1],[II.2], and [III.7] which introduce complexity theory to social choice and prove several novel theorems that complement the famous Arrow theorem on the impossibility of fair voting schemes. In [II.1] it is shown that two historical voting schemes have unfortunate property that it is NP-hard to tell whether any particular candidate has won the election. (One of these voting schemes is due to Lewis Carroll, who seriously suggested it for Oxford University; we think he would have liked this result.) We also prove an "impracticality" theorem, which is a computational analogue of Arrow's theorem. Our theorem says any voting rule that does enough work to meet certain minimal conditions of fairness must do so much work that it is impractical; that is, it is NP-hard to determine who won the election! In [II.2] we discuss another famous impossibility theorem due to Gibbard and Satterthwaite; it says that any voting scheme that meets certain minimal conditions of fairness must be susceptible to manipulation by strategic voting. We give an algorithm that will manipulate most commonly-used voting schemes in polynomial-time. More interestingly, we show a real voting rule that is in principle susceptible to manipulation by strategic voting, but is computationally resistant; that is, it is NP-hard to determine how to manipulate! This suggests that computational complexity can protect the integrity of social choice. Even where abuse is possible, it might not be practical on computational grounds. (This is similar to current ideas in cryptography, where an ideal cryptographic scheme is easy to use but computationally hard to decrypt.) In [III.7] we study the computational complexity of several other forms of manipulating an election, and show how different voting schemes vary in their susceptibility. The main result is that, while Arrow's theorem says that four rationality criteria are inconsistent, for some voting schemes it is NP-hard to recognize inconsistencies! We expect that these three papers will have a considerable impact on social choice theory.

I. REFEREED PUBLICATIONS

(1) "Matching Based Interactive Facility Layout," IEE Transactions, IIE Transactions, Vol. 19, No. 3, September 1987.

(2) "An Implicit Enumeration Procedure for the General Linear Complementarity Problem," Mathematical Programming Study, 31, pp. 1987.

(3) "Solving Large Scale Linear Programs by Aggregation," Computers and Operations Research, Vol. 14, No. 5, 1987.

(4) "Order Picking in An Aisle," IIE Transactions, Vol. 20, No. 1, March 1988.

(5) "An Efficient Algorithm to Cluster Order Picking Items in a Wide Aisle," Engineering Costs and Production Economics, Vol 13, 1988.

(6) "Sequencing Picking Operations for a Man-Aboard Order Picking System, Material Flow, Vol. 4, No. 4, March 1988.

(7) "Design of efficient bin-numbering schemes for warehouses", Material Flow, Vol. 4, No. 4, March 1988.

(8) "A simple and efficient algorithm to compute tail probabilities from transforms", Operations Research, Vol 36, No. 1, 1988.

(9) "A Decomposition Procedure for Convex Quadratic Programs," Naval Research Logistics Quarterly, Vol. 35, 1988.

(10) "Heuristics based on spacefilling curves for combinatorial problems in Euclidean space", Management Science, Vol.34, March 1988.

## II. PAPERS ACCEPTED FOR PUBLICATION

(1) "Voting schemes for which it can be difficult to tell who won the election", to appear in Social Choice and Welfare.

(2) "The computational difficulty of manipulating an election", to appear in Social Choice and Welfare.

(3) "Decentralized control of automated guided vehicles on a simple loop", to appear in IIE Transactions.

(4) "Spacefilling curves and the travelling salesman problem", to appear in J. Assoc Comput Mach.

(5) "Expected performance of shelf heuristics for 2-dimensional packing", to appear in OR Letters.

(6) "Optimizing the Location of Input/Output Stations within Facilities Layout," to appear in Material Flow.

(7) "On a Node Ranking Problem of Trees and Graphs," to appear in Information Processing Letters.

(8) "Utilizing Cut Trees as Design Skeletons for Facility Layout," to appear in IIE Transactions.

(9) "Shared Storage Policies Based on Duration of Stay," to appear in Management Science.

III. PAPERS IN REFEREEING

(1) "Interactive Optimization Methodology for Fleet Scheduling," NLRQ.

(2) "Hierarchical Solution of Network Flow Problems" Networks.

(3) "Determining Optimal Lane Depths for Single and Multiple Products in Block Stacking Operations," IIE Transactions.

(4) "Location Issues in Guaranteed Time Distribution Systems", Management Science.

(5) "On an Edge Ranking Problem of Trees and Graphs," Discrete Applied Mathematics.

(6) "Parallel Assembly of Modular Products - An Analysis," European Journal of Operations Research.

(7) "How hard is it to cheat in an election?", Econometrica.

## IV. TECHNICAL REPORTS

Ratliff, H. D., and M. Goetschalckx, "An Efficient Algorithm to Cluster Order Picking Items in a Wide Aisle," PDRC Report 86-12, ISyE Department, Georgia Institute of Technology.

Ratliff, H. D., and B. Montreuil, "Utilizing Cut Trees as Design Skeletons for Facility Layout," PDRC Report 87-02, ISyE Department, Georgia Institute of Technology.

Ratliff, H. D., Iyer, A., and G. Vijayan, "On a Node Ranking Problem of Trees and Graphs," PDRC Report 87-03, ISyE Department, Georgia Institute of Technology.

Ratliff, H. D., and M. Goetschalckx, "Optimal Lane Depths for Single and Multiple Products in Block Stacking Storage Systems," PDRC Report 87-04, ISyE Department, Georgia Institute of Technology.

Ratliff, H. D. and B. Montreuil, "Optimizing the Locations of Input/Output Stations Within Facilities Layout," PDRC Report 87-05, ISyE Department, Georgia Institute of Technology.

Iyer, A. V., J. J. Jarvis, and H. D. Ratliff, "An Averaging Algorithm for MODES," PDRC Report No. 87-06, Georgia Institute of Technology, 1987.

Iyer, A. V., J. J. Jarvis, and H. D. Ratliff, "Hierarchical Approaches to Solution of MODES," PDRC Report No. 87-07, Georgia Institute of Technology, 1987.

Iyer, A. V. and Ratliff H. D., "Location Issues in Guaranteed Time Distribution Systems," PDRC Report No. 87-08, Georgia Institute of Technology, 1987.

Al-Khayyal, F. A. and Kyparisis, J., "Conditions for Finite Convergence of Algorithms for Nonlinear Programs and Variational Inequalities" PDRC Report No. 88-01, Georgia Institute of Technology, 1988.

Al-Khayyal, F., Horst, R. and Pardalos, P. M., "Global Optimization of Concave Functions Subject to Separable Quadratic Constraints and of All-Quadratic Separable Problems," PDRC Report No. 88-04, Georgia Institute of Technology, 1988.

Iyer, A. V., Ratliff, H. D. and Vijayan, G., "Parallel Assembly of Modular Products: An Analysis," PDRC Report No. 88-07, Georgia Institute of Technology, 1988.

Iyer, A. V., Ratliff, H. D. and Vijayan, G. "On An Edge Ranking Problem of Trees and Graphs," PDRC Report No. 88-08, Georgia Institute of Technology, 1988.

## V. INVITED PRESENTATIONS

"Implementation Issues in Microcomputer Vehicle Routing," TIMS/ORSA Joint National Meeting, New Orleans, May 1987.

"Graph Theory Based Assembly Planning," TIMS/ORSA Joint National Meeting, New Orleans, May 1987.

"Location Issues in Guaranteed Time Distribution Systems," ORSA/TIMS Joint National Meeting, St. Louis, October 1988.

"Optimization Issues in Deployment Planning," EURO IX, TIMS XXVII, AFCET Joint International Meeting, Paris-France, July 1988.

Table Ronde sur l'Analyse de l'Aggregation des Preferences et des Choix en l'Honneur de Marquis de Condorcet, Centres International des Rencontres Mathematiques, Marseille-Luminy; "The computational difficulty of manipulating an election", April 1988.

Yale University, Graduate Seminar of the School of Organization and Management; "The computational complexity of social choice", November 1987.

Yale University, "The Complexity of Voting," November 1987.

The University of Florida, Graduate Seminar of the Department of Industrial and Systems Engineering: "The computational complexity of social choice", October 1987.

University of Florida, "The Complexity of Voting," October 1987.

"Matrix Classes in Linear Complementarity Theory Characterized by Solutions to Linear Programs," ORSA/TIMS Joint National Meeting, New Orleans, Louisiana, May 1987.

"Solving Linear Complementarity Problems as Linear Programs Revisited," SIAM Conference on Optimization, Houston, Texas, May 1987.

"The Role of Interactive Optimization in Research," ORSA/TIMS Joint National Meeting, New Orleans, May 1987, abstract published in ORSA/TIMS Bulletin, No. 23, 1987.

"Implementation Issues in Microcomputer Vehicle Routing," ORSA/TIMS Joint National Meeting, New Orleans, May 1987, abstract published in ORSA/TIMS Bulletin, No. 23, 1987.

"Graph Theory Based Assembly Planning," ORSA/TIMS Joint National Meeting, New Orleans, May 1987.

Massachusetts Institute of Technology, Graduate Seminar in Operations Research; "The computational complexity of social choice methods", April 1987.

The University of Arizona, Graduate Seminar in Decision Sciences; "The computational complexity of social choice methods", April 1987.

The University of Arizona, "The complexity of voting," April 1987.

Massachusetts Institute of Technology, "The Complexity of Voting," April 1987.

"Industry-University Cooperation: The TIMS Faculty-in-Residence Program," Panel discussion, ORSA/TIMS St. Louis Meeting; abstract published in ORSA/TIMS Bulletin, No. 24, 1987.

"Experience with Outside\In Routing Procedures," IFORS 11th Triennial Conference on Operations Research, Buenos Aires, Argentina, 1987.

"A Microcomputer Logistics Workstation," ORSA/TIMS Joint National Meeting, Miami, October 1986.

"Exploiting Pure Network Substructure in Generalized Networks," ORSA/TIMS Joint National Meeting, Miami, October 1986.

"Warehouse Layout Design with Block Stacking Storage," ORSA/TIMS Joint National Meeting, Miami, October 1986.

## VI. OTHER

Dr. H. Donald Ratliff is serving as Editor in Chief of the Journal of Operations Research.

Dr. John J. Bartholdi is serving as Departmental Editor for New or Non-traditional applications for the Journal of Operations Research and as Departmental Editor for Planning, Scheduling, and Control for the IIE Transactions.

A new method has been developed for solving matching problems. This work was the runner-up in the Nicholson paper completion.

The paper "Retrieval strategies for a carousel conveyor", by J. J. Bartholdi III and L. K. Platzman was selected as the best paper to appear in IIE Transactions during 1986.

"Hierarchical Solutions of Network Flow Problems,"
ORSA/TIMS Joint National Meeting, Miami, October 1986.